**Deploying Containers on Windows Server**

First up, we’re going to use the Azure CLI to deploy a Windows 2016 Server to Azure. After this is deployed we will follow the manual steps to setup Windows Containers and Docker.

1. Have a look at the [WindowsVirtualMachine.json](https://raw.githubusercontent.com/MTCCloudInfrastructure/contosony-workshop-templates/master/lab01/101-simple-windows-vm/ws2016.json) file that we will be deploying.

**Create Resource Group**

A resource group is a grouping of Azure resources that can be managed and secured as a single unit. Read "I can delete every resource in a resource group by deleting the resource group"... kind of scary.

**Create a resource group from the Azure-CLI:**

*azure group create {RESOURCE GROUP NAME} southcentralus*

Replace {RESOURCE GROUP NAME} with whatever you like. The "southcentralus" at the end is the data center location. There something like 22+ DCs now...

**Deploy the VM**

Now it's time to create a VM...

**Deploy an ARM Template using the Azure-CLI:**

*azure group deployment create {RESOURCE GROUP NAME} {DEPLOYMENT NAME --template-uri* [*https://raw.githubusercontent.com/MTCCloudInfrastructure/contosony-workshop-templates/master/lab01/101-simple-windows-vm/ws2016.json*](https://raw.githubusercontent.com/MTCCloudInfrastructure/contosony-workshop-templates/master/lab01/101-simple-windows-vm/ws2016.json)

Replace {RESOURCE GROUP NAME} with the resource group name you just created.

Replace {DEPLOYMENT NAME} with whatever you like.

This command creates a deployment with the resource manager and passes the URI of the Windows template we just reviewed. It will also prompt you for the following parameters:

1. Username (don't use "admin")
2. Password (needs to be more than 12 chars and be complex)
3. DNS Label (this will be the DNS prefix used to connect to the box)

**Remote Desktop to your new Windows Box**

From the start menu open Microsoft Remote Desktop

*DNS-LABLE-YOU-CREATED.eastus.cloudapp.azure.com*

**The below content is preliminary content and subject to change.**

This exercise will walk through basic deployment and use of the Windows container feature on Windows Server. After completion, you will have installed the container role and have deployed a simple Windows Server container. Before starting this quick start, familiarize yourself with basic container concepts and terminology.

From the VM we've just created in Azure:

**1. Install Container Feature**

The container feature needs to be enabled before working with Windows containers. To do so run the following command in an elevated PowerShell session.

*Install-WindowsFeature containers*

When the feature installation has completed, reboot the computer.

*Restart-Computer -Force*

**2. Install Docker**

Docker is required in order to work with Windows containers. Docker consists of the Docker Engine, and the Docker client. For this exercise, both will be installed.

Download and install the Windows-native Docker Engine

*Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force*

*Install-Module -Name DockerMsftProvider -Force*

*Install-Package -Name docker -ProviderName DockerMsftProvider -Force*

*Restart-Computer -Force*

Docker Engine is now installed and running as service. To verify Docker is running, type:

*docker version*

**3. Install Base Container Images**

Windows containers are deployed from templates or images. Before a container can be deployed, a base OS image needs to be downloaded. To search Docker Hub for Windows container images, run *docker search Microsoft.*

*docker search microsoft*

*NAME DESCRIPTION*

*microsoft/aspnet ASP.NET is an open source server-side Web ...*

*microsoft/dotnet Official images for working with .NET Core...*

*mono Mono is an open source implementation of M...*

*microsoft/azure-cli Docker image for Microsoft Azure Command L...*

*microsoft/iis Internet Information Services (IIS) instal...*

*microsoft/mssql-server-2014-express-windows Microsoft SQL Server 2014 Express installe...*

*microsoft/nanoserver Nano Server base OS image for Windows cont...*

*microsoft/windowsservercore Windows Server Core base OS image for Wind...*

*microsoft/oms Monitor your containers using the Operatio...*

*microsoft/dotnet-preview Preview bits for microsoft/dotnet image*

*microsoft/dotnet35*

*microsoft/applicationinsights Application Insights for Docker helps you ...*

*microsoft/sample-redis Redis installed in Windows Server Core and...*

*microsoft/sample-node Node installed in a Nano Server based cont...*

*microsoft/sample-nginx Nginx installed in Windows Server Core and...*

*microsoft/sample-httpd Apache httpd installed in Windows Server C...*

*microsoft/sample-dotnet .NET Core running in a Nano Server container*

*microsoft/sqlite SQLite installed in a Windows Server Core ...*

*...*

The following command will download the Windows Server Core base image.

*docker pull microsoft/windowsservercore*

This process can take some time, so take a break and pick back up once the pull has completed.

Once the image is pulled, running docker images will return a list of installed images, in this case the Windows Server Core image.

*docker images*

*REPOSITORY TAG IMAGE ID CREATED SIZE*

*microsoft/windowsservercore latest 02cb7f65d61b 8 weeks ago 7.764 GB*

**4. Deploy Your First Container**

For this exercise, you will download a pre-created IIS image from the Docker Hub registry and deploy a simple container running IIS.  
Download the IIS image using docker pull.

*docker pull microsoft/iis*

The image download can be verified with the *docker images* command. Notice here that you will see both the base image (windowsservercore) and the IIS image.

*docker images*

*REPOSITORY TAG IMAGE ID CREATED SIZE*

*microsoft/iis latest accd044753c1 11 days ago 7.907 GB*

*microsoft/windowsservercore latest 02cb7f65d61b 8 weeks ago 7.764 GB*

*User docker run to deploy the IIS container.*

*docker run -d -p 80:80 microsoft/iis ping -t localhost*

This command runs the IIS image as a background service (-d) and configures networking such that port 80 of the container host is mapped to port 80 of the container. For in depth information on the Docker Run command, see [Docker Run Reference on Docker.com](https://docs.docker.com/engine/reference/run/).

Running containers can be seen with the docker ps command. Take note of the container name, this will be used in a later step.

*docker ps*

*CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAME*

*09c9cc6e4f83 microsoft/iis "ping -t localhost" About a minute ago Up About a minute 0.0.0.0:80->80/tcp big\_jang*

From a different computer, open up a web browser and enter the IP address of the container host. If everything has been configured correctly, you should see the IIS splash screen. This is being served from the IIS instance hosted in the Windows container.

**Note:** if you are working in Azure, the external IP Address of the virtual machine will be needed to access the IIS website.

Back on the container host, use the docker rm command to remove the container. Note – replace the name of the container in this example with the actual container name.

*docker rm -f big\_jang*

**Delete the Resource Group**

This command will remove everything you just created!

*azure group delete {RESOURCE GROUP NAME} -q*